

# COMPRESSION OF IMAGES IN CFA FORMAT

Halil I. Cuce, A. Enis Cetin

Department of Electrical and Electronics Engineering,  
Bilkent University, Ankara, Turkey  
E-mail: {hic, cetin}@ee.bilkent.edu.tr

Mark K. Davey

Grandeye Ltd. 6 Huxley Road,  
Guildford, Surrey, GU2 7RE, UK  
E-mail: mark.davey@grandeye.com

## ABSTRACT

In this paper, images in Color Filter Array (CFA) format are compressed without converting them to full-RGB color images. Green pixels are extracted from the CFA image data and placed in a rectangular array, and compressed using a transform based method without estimating the corresponding luminance values. In addition, two sets of color difference (or chrominance) coefficients are obtained corresponding to the red and blue pixels of the CFA data and they are also compressed using a transform based method. The proposed method produces better PSNR values compared to the standard approach of bilinear interpolation followed by compression.

**Index Terms**— Image coding, discrete cosine transforms, quantization, Huffman codes

## 1. INTRODUCTION

Most color imaging systems and circuits use a color filter array (CFA) to select different wavelength bands at different photosensor locations. The Bayer filter with a mosaic pattern shown in Fig. 1 is commonly utilized in digital cameras [1, 2]. Cameras using this pattern produce only one color value per pixel. On the other hand, digital image coding techniques and standards, including JPEG and JPEG-2000, were developed for full Red, Green, and Blue (RGB) color images in which each pixel of the image consists of three color components [3, 4, 5]. In order to compress digital images produced by imaging sensors using CFA arrays the standard procedure is to interpolate the CFA image data to the full RGB color format and compress the interpolated full color image using an image compression standard. The interpolation process initially increases the data size three-folds and this not only increases the computational cost but also requires an additional memory space inside the digital camera.

In this paper, images in Color Filter Array (CFA) format are compressed without converting them to full-RGB color images. Green pixels are extracted from the CFA image data and placed in a rectangular array, and compressed using a transform based method without estimating the corresponding luminance values. In addition, two sets of color differ-

$G_{11}$	$R_{12}$	$G_{13}$	$R_{14}$	$G_{15}$
$B_{21}$	$G_{22}$	$B_{23}$	$G_{24}$	$B_{25}$
$G_{31}$	$R_{32}$	$G_{33}$	$R_{34}$	$G_{35}$
$B_{41}$	$G_{42}$	$B_{43}$	$G_{44}$	$B_{45}$
$G_{51}$	$R_{52}$	$G_{53}$	$R_{54}$	$G_{55}$

**Fig. 1.** A 5 by 5 section of a Colour Filter Array (CFA) in Bayer format.  $G_{ij}$ ,  $R_{ij}$  and  $B_{ij}$  represent a green, a red, and a blue pixel, respectively.

ence (or chrominance) coefficients are obtained corresponding to the red and blue pixels of the CFA data and they are also compressed using a transform based method. The main advantage of the proposed method is that it gives more emphasis to green pixels as human vision system [2]. Another advantage is that it does not require additional memory space for interpolation inside a digital camera.

## 2. PROPOSED METHOD

The method described in this article compresses digital images in CFA format without converting them to full-RGB color images. Only one-third of full-color data is available in an image obtained by a sensor using the Bayer filter. In Bayer pattern, there are two green pixels, one red pixel and a blue pixel in a two by two image region as shown in Fig 1. The human vision system gets most of its sharpness information from green light, this is the main reason why there are more green pixels compared to red and blue pixels in Bayer pattern [2]. Instead of interpolating the missing image data we rearrange the available data into other two-dimensional arrays representing the color information and compress the rearranged data. For example, the available green color data can be re-ordered into a (rectangular) two-dimensional array

as follows:

$$[g_{ij}] = \begin{bmatrix} G_{11} & G_{13} & G_{15} & \cdots \\ G_{21} & G_{23} & G_{25} & \cdots \\ G_{31} & G_{33} & G_{35} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (1)$$

Entries of the above matrix obey the following rule:

$$g_{ij} = \begin{cases} G_{i,2j-1} & \text{if } i \text{ is odd, and} \\ G_{i,2j} & \text{if } i \text{ is even} \end{cases} \quad (2)$$

In this way, the green CFA data in the quincunx format in the CFA array is rearranged into a two-dimensional array, and a plurality of square 8 by 8 blocks can be obtained from the two-dimensional array for DCT based block compression purposes. Green pixel values can be also rearranged diagonally as in [6, 7]. This arrangement is discussed in the next section.

In our scheme, the green data is compressed as is without trying to estimate the luminance values as in most coding schemes. Since the human eye gives more emphasis to the green channel information more emphasis should be given to the green channel information. This is one of the main differences between our method and other CFA compression methods [1, 2, 9, 10, 11] which estimate a luminance value as a weighted average of the current green value and neighboring red and blue values. Estimated luminance values are not perfect because the actual red and blue values are not available for green pixels in a CFA array. This is the main reason why we get better compression results than methods estimating the luminance value for green pixels.

Red and blue channel data are encoded using color differencing approach or chrominance data compression. Color difference values are obtained from the two-by-two blocks of the CFA array as well. For example, consider the first two-by-two block of the CFA data containing the samples  $(G_{11}, R_{12}, B_{21}, G_{22})$  of the array shown in Fig. 1. From these four values two color difference values for the red and blue pixels are obtained as follows

$$\begin{aligned} u_{11} &= R_{12} - \frac{G_{11} + G_{22}}{2} \\ v_{11} &= B_{21} - \frac{G_{11} + G_{22}}{2} \end{aligned} \quad (3)$$

respectively. In general, a colour difference value for each red pixel is obtained using the formula:

$$u_{ij} = R_{2i-1,2j} - \frac{G_{2i-1,2j-1} + G_{2i,2j}}{2} \quad (4)$$

and, similarly, a colour difference value for each blue pixel is obtained:

$$v_{ij} = B_{2i,2j-1} - \frac{G_{2i-1,2j-1} + G_{2i,2j}}{2} \quad (5)$$

In this way, the data in Bayer color representation is converted into Green - color difference U - color difference V (GUV)

color space. The size of U and V matrices are half of the G matrix as in 4:2:2 subsampling scheme. After this step, the U and V array data are divided into plurality of blocks and the Discrete Cosine Transform of the blocks are computed. Transform domain data is quantized and Huffman encoded using the quantization tables of the JPEG standard in our simulations studies described in Section 4. Other compression methods and standards including the wavelet based JPEG 2000 standard can be also applied to this rearranged data for compression.

### 3. QUINCUNX SCANNING OF THE CFA DATA

There are other ways of rearranging the green color values of the CFA data. Green color filter of the CFA is in quincunx format. Therefore, the filter array can be rotated 45 degrees and two-dimensional plurality of blocks can be extracted from the diamond shaped partitions of the CFA data. For example, the following two-dimensional (rectangular) array can be obtained from Fig. 1 as follows:

$$[d_{ij}] = \begin{bmatrix} G_{31} & G_{22} & G_{13} & \cdots \\ G_{42} & G_{33} & G_{24} & \cdots \\ G_{53} & G_{44} & G_{35} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (6)$$

The original data actually comes from a diamond-shaped region from the CFA array as shown in Fig. 2. In this way, the green or the corresponding luminance data can be rearranged into two-dimensional plurality of rectangular blocks and compressed by an ordinary Discrete Cosine Transform based image coder. Green color filter of the CFA is in quincunx format with periodicity vectors  $[1 \ 1]^T$  and  $[1 \ -1]^T$ . In other words, these vectors describe the quincunx array:  $Q = \{[n \ k]^T \mid n[1 \ 1]^T + k[1 \ -1]^T\}$ . The array Q includes index pairs  $[n, k] = [0, 0], [1, 1], [1, -1], [2, 0], [2, 2], [1, 3], [3, 3], \dots$  of a two-dimensional array but it does not include all pairs of indices of a two-dimensional array for example  $[0, 1], [1, 0], [2, 1], \dots$  etc. The periodicity matrix describing the quincunx array is formed from the periodicity vectors:

$$P = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad (7)$$

This matrix can be diagonalized using the Smith-normal form as described in the article by Gunduzhan, Cetin and Tekalp [6]. Diagonalization process defines a variety of means of mapping the quincunx array into a two-dimensional (rectangular) array. In the case of a quincunx array, the relation defined by the matrix  $P^T$  maps the quincunx array into a two-dimensional array.

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} \quad (8)$$

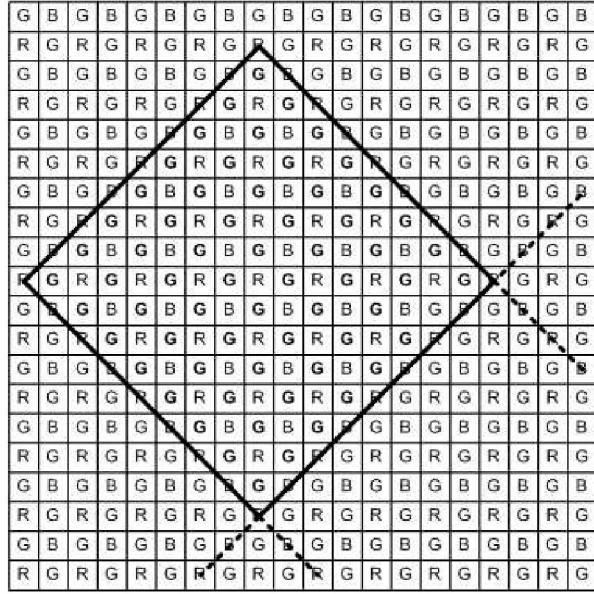


Fig. 2. The method to extract 8x8 blocks for Green values from CFA image.

As a result, the following rule is a way to map the quincunx array into a rectangular two-dimensional array,

$$g[i, j] = G[i + j, i - j] \quad (9)$$

The re-ordered data  $g[i, j]$  covers all the indices of a two-dimensional array, i.e.,  $g[0, 0] = G[0, 0]$ ,  $g[1, 0] = G[1, 1]$ ,  $g[1, 1] = G[2, 0]$ ,  $g[0, 1] = G[1, -1]$ , .... Once the original quincunx data in the CFA array is in the form of a two-dimensional array then the data can be divided into plurality of blocks and transformed into compression domain using the DCT.

#### 4. SIMULATION EXAMPLES

We test the proposed CFA image compression method using some well-known images: Lena, peppers, monarch, girl, by assuming the availability of only one color value for each pixel in the CFA format. Also, we use some other images: lighthouse, a widely used image to test the quality of CFA demosaicking algorithms [8] and an image from the Halocam camera of Grandeye Ltd [12].

In order to compare the performance of the proposed method with JPEG, Peak Signal to Noise Ratio (PSNR) measure is used. The PSNR is computed as follows:

$$MSE = \frac{1}{w \times h} \sum_y \sum_x [O(x, y) - R(x, y)]^2 \quad (10)$$

$$PSNR \text{ (in dB)} = 20 \log_{10} \frac{255}{\sqrt{MSE}}$$

where  $O$  and  $R$  are the original and the reconstructed images, respectively.

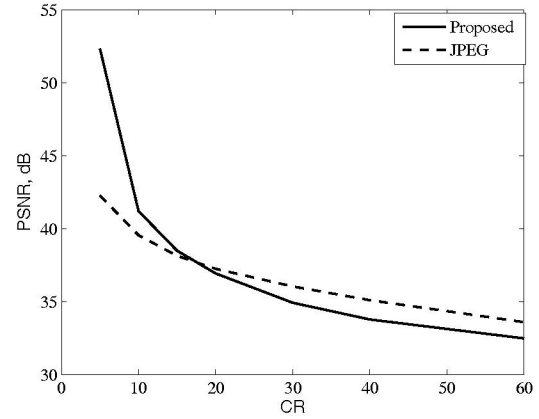


Fig. 3. Average PSNR values for six test images for the proposed method and the JPEG compression.

Table 1. PSNR values of test images for various CR values.

CR	Girl		Lena		Monarch	
	Jpeg	Ours	Jpeg	Ours	Jpeg	Ours
5	39.37	46.96	40.02	49.89	39.78	50.15
10	36.58	38.85	37.24	37.80	37.49	41.22
15	34.96	35.23	35.07	34.97	35.85	37.06
20	33.97	33.49	34.11	33.45	34.60	34.33
30	32.61	31.68	33.11	31.46	32.90	30.98
40	31.59	30.63	31.84	30.34	31.49	29.41
60	29.91	29.54	30.12	29.07	29.30	27.72

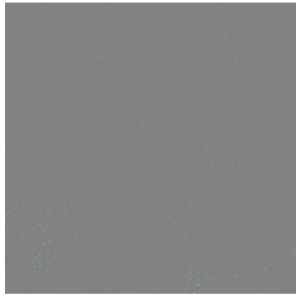
CR	Lighthouse		Peppers		Grandeye	
	Jpeg	Ours	Jpeg	Ours	Jpeg	Ours
5	35.23	45.98	38.56	46.91	44.62	55.48
10	33.28	36.65	35.23	36.23	41.79	43.11
15	31.96	32.32	33.72	33.67	40.72	41.09
20	30.77	30.31	32.74	32.12	40.13	40.12
30	29.43	28.22	31.49	30.40	39.16	38.78
40	28.47	27.19	30.50	29.39	38.70	37.89
60	27.05	25.82	29.02	28.10	37.79	36.97

Average PSNR values versus Compression Ratio (CR) values for these six images are shown in Fig. 3. GUV planes of the images are compressed using the DCT and Huffman tables of the JPEG standard. These images are also interpolated to RGB images using bilinear interpolation and JPEG compressed. The JPEG compression results are also plotted in Fig. 3. PSNR values of individual test images for various CR values are given in Table 1. The proposed method produces better PSNR when the compression ratio is below CR=15 compared to the standard approach of bilinear interpolation followed by the JPEG compression. After CR=15 one can notice severe compression artifacts in most natural images in JPEG standard. Because of this higher compression ratios are rarely used in practice.

Fig. 4 shows difference images between the original images and the images decompressed by the proposed method



(a)



(b)



(c)

**Fig. 4.** A portion from the lighthouse image. (a) Original. (b) and (c) Differences between original image and images reconstructed by the proposed method and JPEG at CR=5, respectively. Difference image= $128+4*\text{abs}(\text{original} - \text{reconstructed})$ .

and JPEG in order to show the improvement of our method since human eye system may not distinguish any difference when the actual reconstructed images are depicted. The difference images are multiplied by 4 to make them more visible. The edges in the proposed method are sharper than the edges in JPEG as shown in Fig. 4-b and 4-c.

Although only one case from omnidirectional Grandeye camera images is reported in this section, extensive simulation studies were carried out using Grandeye camera images and the proposed method outperformed the standard approach in all cases.

## 5. CONCLUSION

In this paper, a compression scheme for CFA image data is presented. The proposed method has better performance than JPEG when compression ratio is below 15 in our simulation studies. The proposed method has also low computational cost and requires low memory when compared to the standard methods using full RGB values since the CFA image data is one-third of the RGB image data.

## 6. REFERENCES

- [1] Y. Tsai, K. T. Parulski, M. A. Rabbani, "Compression method and apparatus for single-sensor color imaging systems," US Patent No. 5,065,229, Oct. 1, 1991.
- [2] <http://www.kodak.com/US/en/corp/researchDevelopment/technology/Features/pixPic.shtml>
- [3] W. Pennebaker and J. Mitchell, "JPEG: still image data compression standard," Van Nostrand Reinhold, NY, 1992.
- [4] M. Rabbani, Rajan Joshi, "An overview of the JPEG2000 still image compression standard," *Signal Processing: Image Communication*, vol. 17, pp. 3-48, 2002.
- [5] JPEG 2000 Image Coding System, ISO/IEC International Standard, 5444-1, 2000.
- [6] E. Gunduzhan, A. Enis Cetin, A. Murat Tekalp, "DCT Coding of Nonrectangularly Sampled Images," *IEEE Signal Processing Letters*, vol. 1, no. 9, pp. 131-134, Sept. 1994.
- [7] R. Ansari, A. E. Cetin and S. H. Lee, "Subband Coding of Images using Nonrectangular Filter Banks," in *Proc. of the SPIE 32nd Annual International Technical Symposium: Applications of Digital Signal Processing*, San Diego, CA, vol. 974, August 1988.
- [8] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: Color filter array interpolation," *IEEE Signal Processing Magazine*, pp. 44-54, Jan. 2005.
- [9] S. Battiato, A. Buemi, L. Della Torre, A. Vitali, "Fast Vector Quantization Engine for CFA Data Compression," in *Proc. of IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, NSIP-2003 Grado, Italy, June 2003.
- [10] Sang-Yong Lee, Ortega, A, "A novel approach of image compression in digital cameras with a Bayer color filter array," *Proc. of IEEE International Conference on Image Proc.*, vol.3, pp. 482-485, 2001.
- [11] Koh, C.C. Mitra, S.K., "Compression of Bayer color filter array data," *Proceedings of IEEE International Conference on Image Processing*, vol. 2, pp. II-255-8, 14-17 Sept. 2003.
- [12] <http://www.grandeye.com>